CLAIMS

What is claimed is:

1. A system of securely using decryption keys during Programmable Logic Device (PLD) configuration, comprises:

a microcontroller for receiving an encrypted bitstream;

a key storage register coupled to the microcontroller for storing key data;

a decryptor coupled to the key storage register, wherein only the decryptor can read from the key storage register; and

a configuration data register in the PLD, wherein the configuration data register cannot be read by the microcontroller after the decryptor is used.

2. The system of claim 1 wherein the microcontroller stores key data in the key storage register, but the microcontroller cannot read from the key storage register.

3. The system of claim 2, wherein the decryptor is a hardware decryptor embedded in an integrated circuit along with the PLD.

4. The system of claim 1, wherein the decryptor is a software decryptor stored in a memory that uses hardware to enable access to the key storage register based on a memory address.

5. The system of claim 4, wherein the memory is a ROM having a decryption engine.

6. The system of claim 1, wherein the microcontroller further receives a configuration boot program along with the encrypted bitstream.

7. The system of claim 1, wherein the microcontroller, the key register, the decryptor, and the configuration data register are all within the PLD.

8. The system of claim 1, wherein the microcontroller is an emulated microcontroller in the PLD.

9. A system of securely using decryption keys during configuration of an integrated circuit having programmable logic, comprising:

a microcontroller for receiving an encrypted bitstream;

a key storage register coupled to the microcontroller for storing key data;

a decryption program stored in a memory that uses a predetermined memory address to enable access to the key storage register; and

a configuration data register in the FPGA, wherein the configuration data register cannot be read by the microcontroller after the decryption programmed is used.

10. The system of claim 9, wherein the memory is a ROM containing a decryption engine.

11. The system of claim 9, wherein the microcontroller further receives a configuration boot program along with the encrypted bitstream.

12. A method of securely using decryption keys during field programmable gate array configuration, comprising the steps of:

receiving an encrypted bitstream at a microcontroller;

loading a decryptor with data from a key register;

loading the decryptor with data from the microcontroller; and

loading a configuration data register with a decrypted bitstream from the decryptor, wherein the configuration data register cannot be read by the microcontroller after the decryptor is used.

13. The method of claim 12, wherein the method further comprises the step of loading the key register with key data from the microcontroller.

14. The method of claim 12, wherein the microcontroller can only be read by the configuration data register after the decryptor is used.

15. The method of claim 12, wherein the microcontroller cannot read from the key register.

16. The method of claim 12, wherein only the decryptor can read from the key storage register.

17. The method of claim 12, wherein the steps of loading the decryptor with data from key register and loading the decryptor with data from the microcontroller comprises using a predetermined instruction enabling access to the key storage register based on a known address of a memory storing a decryption engine forming the decryptor.

18. A system of securely using decryption keys during programmable logic device configuration, comprises:

a memory-mapped key register coupled to a microcontroller data bus;

a decryptor engine stored in non-volatile memory and coupled to the microcontroller data bus; and

logic circuitry limiting access to the key register from the microcontroller data bus using specified addresses of the non-volatile memory.

19. The system of claim 18, wherein the logic circuitry uses specified addresses of the non-volatile memory by limiting access to minimum and maximum ROM memory addresses using a microcontroller program counter.

20. A bitstream, comprising:

a configuration boot program for running a microcontroller on a programmable logic device; and

an encrypted bitstream portion of the bitstream containing encrypted configuration data for a configuration data register on the programmable logic device.

21. The bitstream of claim 20, wherein said configuration boot program comprises instructions for a decryptor.

22. The bitstream of claim 20, wherein said configuration boot program comprises instructions for a decompressor.